

## Research Article

# On Solving System of Linear Differential-Algebraic Equations Using Reduction Algorithm

Srinivasarao Thota 

Department of Applied Mathematics, School of Applied Natural Sciences, Adama Science and Technology University,  
Post Box No. 1888, Adama, Ethiopia

Correspondence should be addressed to Srinivasarao Thota; [srinivasarao.thota@astu.edu.et](mailto:srinivasarao.thota@astu.edu.et)

Received 21 October 2020; Revised 27 November 2020; Accepted 3 December 2020; Published 16 December 2020

Academic Editor: Remi Léandre

Copyright © 2020 Srinivasarao Thota. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we present a new reduction algorithm for solving system of linear differential-algebraic equations with power series coefficients. In the proposed algorithm, we transform the given system of differential-algebraic equations into another simple equivalent system using the elementary algebraic techniques. This algorithm would help to implement the manual calculations in commercial packages such as Mathematica, Maple, MATLAB, Singular, and Scilab. Maple implementation of the proposed algorithm is discussed, and sample computations are presented to illustrate the proposed algorithm.

## 1. Introduction

In many applications of science and engineering, for example, simulation of electric circuits [1–4], mechanical systems [5, 6], and chemical reactions subject to invariants [7–13], the systems of differential-algebraic equations (DAEs) arise naturally, and these systems of DAEs consist of algebraic equations and differential operations. Many engineers and scientists have studied the system of DAEs from a theoretical as well as a numerical point of view and created many new approaches to solve the system of linear differential-algebraic equations; see, for example, [14–25].

In this paper, we are concerned with a linear system of differential-algebraic equations of the following form:

$$\mathcal{A}(x)\partial u(x) + \mathcal{B}(x)u(x) = f(x), \quad (1)$$

where  $x$  is a complex variable,  $\mathcal{A}(x)$  and  $\mathcal{B}(x)$  are  $m \times n$  matrices with analytic functions entries,  $f(x)$  is an  $m$ -dimensional column matrix with analytic functions entries,  $u(x)$  is an  $n$ -dimensional unknown column matrix which is going to be determined, and  $\partial = (d/dx)$  is a differential operator. In this paper, we focus on creating a new reduction algorithm using elementary algebraic techniques as well as the implementation of the proposed algorithm in Maple. Using this algorithm, we can transform a given system of

DAEs into another equivalent system, where we can solve the reduced system easily.

The rest of the paper is organized as follows: in Section 2, we present a new reduction algorithm to solve the given system of DAEs with certain examples to illustrate the proposed reduction algorithm and Section 3 discusses the Maple implementation of the proposed algorithm with sample computations.

## 2. A New Reduction Algorithm

Let  $\mathbb{K}$  be a subfield of the field of complex numbers  $\mathbb{C}$ . Note that  $(\mathbb{Q} \subseteq \mathbb{K} \subseteq \mathbb{C})$ . We denote the ring of formal power series by  $\mathbb{K}[[x]]$  in the variable  $x$  and  $\mathbb{K}((x))$  denotes its quotient field, that is,  $\mathbb{K}((x)) = \mathbb{K}[[x]][x^{-1}]$ . The ring of differential operators is denoted by  $\mathbb{K}[[x]][\partial]$  with coefficients in  $\mathbb{K}[[x]]$ , that is, the set of finite sums  $\sum a_i \partial^i$  with  $a_i \in \mathbb{K}[[x]]$  is equipped with the addition and the multiplication defined by

$$\begin{aligned} \partial^i \partial^j &= \partial^{i+j}, \quad i, j \in \mathbb{N}, \\ \partial f &= f \partial + \frac{df}{dx}, \end{aligned} \quad (2)$$

where  $f \in \mathbb{K}[[x]]$ . Recall the system of DAEs in equation (1):

$$\mathcal{A}(x)\partial u(x) + \mathcal{B}(x)u(x) = f(x), \tag{3}$$

where  $\mathcal{A}(x), \mathcal{B}(x) \in \mathbb{K}[[x]]^{m \times n}$ ,  $f(x) \in \mathbb{K}[[x]]^m$ , and  $u(x) \in \mathbb{K}[[x]]^n$ . The corresponding matrix differential operator of system (3) is  $L = \mathcal{A}\partial + \mathcal{B} \in \mathbb{K}[[x]][\partial]^{m \times n}$ . We recall the basic concepts of the matrix differential operators; see [23, 26–29] for further details.

*Definition 1.* A matrix differential operator  $T \in \mathbb{K}[[x]][\partial]^{m \times n}$  is said to be unimodular matrix if there exists  $V \in \mathbb{K}[[x]][\partial]^{m \times n}$  such that  $VT = TV = I_n$ .

*Definition 2.* Two matrix differential operators  $L, \tilde{L} \in \mathbb{K}[[x]][\partial]^{m \times n}$  are said to be equivalent if there exist two unimodular matrices  $S \in \mathbb{K}[[x]][\partial]^{m \times m}$  and  $T \in \mathbb{K}[[x]][\partial]^{n \times n}$  such that  $\tilde{L} = SLT$ .

The following section presents a new reduction algorithm using elementary algebraic techniques.

*2.1. Reduction Algorithm.* The following lemma is one of the essential steps to create a new reduction algorithm. It shows that any matrix of formal power series centered at origin can be transformed into a block matrix.

**Lemma 1** (see [23, 26, 29]). *Let  $\mathcal{A} \in \mathbb{K}[[x]]^{n \times n}$ . Then there exist two unimodular matrices  $S, T \in \mathbb{K}[[x]]^{n \times n}$  such that*

$$\begin{aligned} S\mathcal{A} &= \begin{pmatrix} \mathcal{A}^{11} \\ 0 \end{pmatrix}, \mathcal{A}T = \begin{pmatrix} \mathcal{A}^{11} & 0 \end{pmatrix}, \\ S\mathcal{A}T &= \begin{pmatrix} \mathcal{A}^{11} & 0 \\ 0 & 0 \end{pmatrix}, \end{aligned} \tag{4}$$

where  $\mathcal{A}^{ij}$  denote the  $i$ -th row and  $j$ -th column block of matrix  $\mathcal{A}$ ,  $\mathcal{A}^{11} \in \mathbb{K}[[x]]^{r \times r}$  is a block matrix, and  $r$  is the rank of matrix  $\mathcal{A}$ .

Suppose that  $L \in \mathbb{K}[[x]][\partial]^{m \times n}$  is a given matrix differential operator. Using Lemma 1, we can construct two unimodular matrix differential operators  $S_1$  and  $T_1$ , by finding the basis of left null space and right null space of the matrix differential operator  $L$ , such that

$$L_1 = S_1 L T_1 = \mathcal{A}_1 \partial + \mathcal{B}_1, \tag{5}$$

where  $\mathcal{A}_1 = \begin{pmatrix} \mathcal{A}^{11} & 0 \\ 0 & 0 \end{pmatrix}$  and  $\mathcal{B}_1 = \begin{pmatrix} \mathcal{B}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ . Now using Lemma 1 to matrix  $\mathcal{A}_1$  of (5), we can get an unimodular matrix  $S_2$  such that

$$L_2 = S_2 L_1 = \mathcal{A}_2 \partial + \mathcal{B}_2, \tag{6}$$

where  $\mathcal{A}_2 = S_2 \mathcal{A}_1 = \begin{pmatrix} \tilde{\mathcal{A}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$  and  $\tilde{\mathcal{A}}^{11}$  is invertible matrix, and  $\mathcal{B}_2 = S_2 \mathcal{B}_1 = \begin{pmatrix} \tilde{\mathcal{B}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ . Again, using Lemma 1 to matrix  $\mathcal{B}_2$  of (6), we can construct an unimodular matrix  $T_2$  such that

$$L_3 = L_2 T_2 = \mathcal{A}_3 \partial + \mathcal{B}_3, \tag{7}$$

where  $\mathcal{A}_3 = \mathcal{A}_2 T_2 = \begin{pmatrix} \tilde{\mathcal{A}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$  and  $\mathcal{B}_3 = \mathcal{B}_2 T_2 = \begin{pmatrix} \tilde{\mathcal{B}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ . If we denote  $S = S_2 S_1$  and  $T = T_1 T_2$ , then we have two unimodular matrix differential operators  $S \in GL_m(\mathbb{K}[[x]])$  and  $T \in GL_n(\mathbb{K}[[x]])$  such that the given system is in reduced form as follows:

$$\tilde{L} = \tilde{\mathcal{A}}\partial + \tilde{\mathcal{B}}, \tag{8}$$

where  $\tilde{L} = SLT$  has the form  $\begin{pmatrix} \tilde{L}^{11} & 0 \\ 0 & 0 \end{pmatrix}$  and  $\tilde{L}^{11}$  is invertible matrix differential operator;  $\tilde{\mathcal{A}}$  has the form  $\begin{pmatrix} \tilde{\mathcal{A}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ ; and  $\tilde{\mathcal{B}}$  has the form  $\begin{pmatrix} \tilde{\mathcal{B}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ . Note that the ranks of  $\tilde{\mathcal{A}}^{11}$  and  $\tilde{\mathcal{B}}^{11}$  may not be the same, but the ranks of  $\tilde{\mathcal{A}}^{11}$  and  $\tilde{\mathcal{B}}^{11}$  are the same only when the coefficient matrices  $\mathcal{A}$  and  $\mathcal{B}$  have the same rank. We generalize the above reduction algorithm in the following theorem.

**Theorem 1.** *Let  $L = \mathcal{A}\partial + \mathcal{B} \in \mathbb{K}[[x]][\partial]^{m \times n}$  be a matrix differential operator. Then we can construct two unimodular matrices  $S \in GL_m(\mathbb{K}[[x]])$  and  $T \in GL_n(\mathbb{K}[[x]])$  such that the given system  $L = \mathcal{A}\partial + \mathcal{B} \in \mathbb{K}[[x]][\partial]^{m \times n}$  is in reduced form:*

$$\tilde{L} = \tilde{\mathcal{A}}\partial + \tilde{\mathcal{B}}, \tag{9}$$

where  $\tilde{L} = SLT$  has the form  $\begin{pmatrix} \tilde{L}^{11} & 0 \\ 0 & 0 \end{pmatrix}$  and  $\tilde{L}^{11}$  is invertible matrix differential operator;  $\tilde{\mathcal{A}}$  has the form  $\begin{pmatrix} \tilde{\mathcal{A}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ ; and  $\tilde{\mathcal{B}}$  has the form  $\begin{pmatrix} \tilde{\mathcal{B}}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ .

*Proof.* Using Lemma 1 to  $L$ , we can construct two unimodular matrices  $S \in GL_m(\mathbb{K}[[x]])$  and  $T \in GL_n(\mathbb{K}[[x]])$ . If we use the substitution  $u(x) = Tv(x)$  in (3) and left multiplying the resultant equation with  $S$ , we have the following reduced form:

$$\begin{aligned} SLTv(x) &= S\mathcal{A}T\partial v(x) + S\mathcal{B}Tv(x) = Sf(x), \\ \text{or } \tilde{L}v &= \tilde{\mathcal{A}}\partial v + \tilde{\mathcal{B}}v = \tilde{f}, \end{aligned} \tag{10}$$

where  $\tilde{L} = SLT \in \mathbb{K}[[x]][\partial]^{m \times n}$ ,  $\tilde{\mathcal{A}} = S\mathcal{A}T \in \mathbb{K}[[x]]^{m \times n}$ ,  $\tilde{\mathcal{B}} = S\mathcal{B}T \in \mathbb{K}[[x]]^{m \times n}$ , and  $\tilde{f} = Sf \in \mathbb{K}[[x]]^{m \times 1}$ .

Indeed, (i) if  $\text{rank}(L) < \text{rank}(\mathcal{A})$  and  $\text{rank}(L) < \text{rank}(\mathcal{B})$ , then the reduced system of DAEs (10) has the following form:

$$\tilde{L}v = \tilde{\mathcal{A}}\partial v + \tilde{\mathcal{B}}v = \tilde{f}, \tag{11}$$

where  $\tilde{L} = \begin{pmatrix} \tilde{L}^{11}\partial & \tilde{L}^{12} & 0 \\ \tilde{L}^{21} & \tilde{L}^{22} & 0 \\ 0 & 0 & 0 \end{pmatrix}$ ,  $v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$ ,  $\tilde{f} = \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \end{pmatrix}$ .

Hence, the system of DAEs in (3) is decomposed into two systems as follows:

$$\begin{aligned} \bar{L}^{11} \partial v_1 + \bar{L}^{12} v_2 \\ \bar{L}^{21} v_1 + \bar{L}^{22} v_2 \end{aligned} \tag{12}$$

with some necessary conditions on the right-hand side expressed by  $\bar{f}_3 = 0$ .

(ii) If  $\text{rank}(L) = \text{rank}(\mathcal{A}) = \text{rank}(\mathcal{B})$ , then the reduced DAS (10) has the following form:

$$\bar{L}v = \bar{\mathcal{A}}\partial v + \bar{\mathcal{B}}v = \bar{f}, \tag{13}$$

where  $\bar{L} = \begin{pmatrix} \bar{L}^{11} & 0 \\ 0 & 0 \end{pmatrix}$ ,  $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ ,  $\bar{f} = \begin{pmatrix} \bar{f}_1 \\ \bar{f}_2 \end{pmatrix}$ . Hence, the system of DAEs in (3) is decomposed as  $\bar{L}^{11} v_1 = \bar{f}_1$  with some necessary conditions on the right-hand side expressed by  $\bar{f}_2 = 0$ .

In the following section, we present certain examples to illustrate the proposed method presented in Theorem 1.  $\square$

### 2.2. Examples

*Example 1.* Consider a matrix differential operator of DAS.

$$\begin{aligned} L &= \mathcal{A}\partial + \mathcal{B} \\ &= \begin{pmatrix} 0 & x & 1-x & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & x & 1-x & 2 & 1 \\ 1 & x-1 & 1 & 0 & 2 \\ 1 & -x & x-1 & 0 & -1 \end{pmatrix} \partial + \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & x+1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & x\partial & (1-x)\partial & \partial+1 & \partial \\ \partial & 0 & -1 & \partial & 0 \\ \partial & x\partial & (1-x)\partial & 2\partial & \partial \\ \partial & (x-1)\partial+x+1 & \partial & 1 & 2\partial \\ \partial & -x\partial & (x-1)\partial & 0 & -\partial \end{pmatrix}. \end{aligned} \tag{14}$$

Using Lemma 1 to  $L$ , one can construct two unimodular matrices  $S_1$  and  $T_1$  (obtained using a basis of left null space and right null space of  $L$ ) as follows:

$$\begin{aligned} \text{left null space of } L &= \left( \frac{2\partial}{\partial-1} \ 0 \ \frac{\partial+1}{1-\partial} \ 0 \ 1 \right), \\ \text{right null space of } L &= \left( \frac{x+1-(x+1)\partial}{(2x-1)\partial+1} \ \frac{1-3\partial}{(2x-1)\partial+1} \ 0 \ \frac{(x+1)\partial-x-1}{(2x-1)\partial+1} \ 1 \right). \end{aligned} \tag{15}$$

Now the unimodular matrices  $S_1$  and  $T_1$  are

$$\begin{aligned} S_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{2\partial}{\partial-1} & 0 & \frac{\partial+1}{1-\partial} & 0 & 1 \end{pmatrix}, \\ T_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 & \frac{x+1-(x+1)\partial}{(2x-1)\partial+1} \\ 0 & 1 & 0 & 0 & \frac{1-3\partial}{(2x-1)\partial+1} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{(x+1)\partial-x-1}{(2x-1)\partial+1} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{16}$$

Thus, multiplying operator  $L$  on the left and right by  $S_1$  and  $T_1$  yields the operator

$$L_1 = S_1 L T_1 = \begin{pmatrix} 1 & x\partial & (1-x)\partial & \partial+1 & \vdots & 0 \\ \partial & 0 & -1 & \partial & \vdots & 0 \\ \partial & x\partial & (1-x)\partial & 2\partial & \vdots & 0 \\ \partial & (x-1)\partial+x+1 & \partial & 1 & \vdots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix}, \tag{17}$$

where

$$\mathcal{A}_1 = \begin{pmatrix} \mathcal{A}^{11} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & x & 1-x & 1 & \vdots & 0 \\ 1 & 0 & 0 & 1 & \vdots & 0 \\ 1 & x & 1-x & 2 & \vdots & 0 \\ 1 & x-1 & 1 & 0 & \vdots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix},$$

$$\mathcal{B}_1 = \begin{pmatrix} \mathcal{B}^{11} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 & \vdots & 0 \\ 0 & 0 & -1 & 0 & \vdots & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 \\ 0 & x+1 & 0 & 1 & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix}. \quad (18)$$

$$S_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (19)$$

Now, using Lemma 1 to matrix  $\mathcal{A}_1$ , we can construct an unimodular matrix  $S_2$ , using a basis of left null space of  $\mathcal{A}_1$ , as follows:

We have

$$L_3 = S_2 L_2 = \mathcal{A}_2 \partial + \mathcal{B}_2 = \begin{pmatrix} 1 & x\partial & (1-x)\partial & \partial+1 & \vdots & 0 \\ \partial & 0 & -1 & \partial & \vdots & 0 \\ \partial & (x-1)\partial+x+1 & \partial & 1 & \vdots & 0 \\ -1 & 0 & 1 & -1 & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix}, \quad (20)$$

where

Again, using Lemma 1 to matrix  $\mathcal{B}_2$  of the matrix differential operator  $L_2$ , we can construct an unimodular matrix  $T_2$  using a basis of right null space of  $\mathcal{B}_2$  as follows:

$$\mathcal{A}_2 = \begin{pmatrix} \tilde{\mathcal{A}}^{11} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & x & 1-x & 1 & \vdots & 0 \\ 1 & 0 & 0 & 1 & \vdots & 0 \\ 1 & x-1 & 1 & 0 & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix},$$

$$\mathcal{B}_2 = \begin{pmatrix} \tilde{\mathcal{B}}^{11} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 & \vdots & 0 \\ 0 & 0 & -1 & 0 & \vdots & 0 \\ 0 & x+1 & 0 & 1 & \vdots & 0 \\ -1 & 0 & 1 & -1 & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix}. \quad (21)$$

$$T_2 = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & \frac{-1}{x+1} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (22)$$

We have

$$L_3 = L_2 T_2 = \mathcal{A}_3 \partial + \mathcal{B}_3 = \begin{pmatrix} 1 & x\partial & (1-x)\partial & \frac{\partial}{1+x} & \vdots & 0 \\ \partial & 0 & -1 & 0 & \vdots & 0 \\ \partial & (x-1)\partial+x+1 & \partial & \frac{-2x\partial}{1+x} & \vdots & 0 \\ -1 & 0 & 1 & 0 & \vdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix}, \quad (23)$$

where

$$\mathcal{A}_3 = \mathcal{A}_2 T_2 = \begin{pmatrix} \tilde{\mathcal{A}}^{11} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & x & 1-x & \frac{1}{x+1} & \vdots & 0 \\ 1 & 0 & 0 & 0 & \vdots & 0 \\ 1 & x-1 & 1 & \frac{-2x}{1+x} & \vdots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix},$$

$$\mathcal{B}_3 = \mathcal{B}_2 T_2 = \begin{pmatrix} \tilde{\mathcal{B}}^{11} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \vdots & 0 \\ 0 & 0 & -1 & 0 & \vdots & 0 \\ 0 & x+1 & 0 & 1 & \vdots & 0 \\ -1 & 0 & 1 & 0 & \vdots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix}. \tag{24}$$

If we denote  $S = S_2 S_1$  and  $T = T_1 T_2$ , then we have two unimodular matrix differential operators:

$$S = S_2 S_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ \frac{2\partial}{\partial-1} & 0 & \frac{\partial+1}{1-\partial} & 0 & 1 \end{pmatrix},$$

$$T = T_1 T_2 = \begin{pmatrix} 1 & 0 & 0 & -1 & \frac{x+1-(x+1)\partial}{(2x-1)\partial+1} \\ 0 & 1 & 0 & \frac{-1}{x+1} & \frac{1-3\partial}{(2x-1)\partial+1} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{(x+1)\partial-x-1}{(2x-1)\partial+1} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{25}$$

We have that the given system is in reduced form as follows:

$$\tilde{L} = \tilde{\mathcal{A}}\partial + \tilde{\mathcal{B}}$$

$$= \begin{pmatrix} 1 & x\partial & (1-x)\partial & \frac{\partial}{1+x} & \vdots & 0 \\ \partial & 0 & -1 & 0 & \vdots & 0 \\ \partial & (x-1)\partial+x+1 & \partial & \frac{-2x\partial}{1+x} & \vdots & 0 \\ -1 & 0 & 1 & 0 & \vdots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{pmatrix}, \tag{26}$$

where  $\tilde{L} = SLT$  has the form  $\begin{pmatrix} \tilde{L}^{11} & 0 \\ 0 & 0 \end{pmatrix}$  and

$$\tilde{L}^{11} = \begin{pmatrix} 1 & x\partial & (1-x)\partial & \frac{\partial}{1+x} \\ \partial & 0 & -1 & 0 \\ \partial & (x-1)\partial+x+1 & \partial & \frac{-2x\partial}{1+x} \\ -1 & 0 & 1 & 0 \end{pmatrix}, \tag{27}$$

is invertible matrix differential operator.

*Example 2.* Consider a matrix differential operator as given below:

$$L = \begin{pmatrix} \partial & (1+x)\partial & 1 & \partial+1 \\ 1 & -x\partial & \partial & -1 \\ \partial+1 & \partial & \partial+1 & \partial \\ -\partial+1 & (-2x-1)\partial & \partial-1 & -\partial-2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & x+1 & 0 & 1 \\ 0 & -x & 1 & 0 \\ 1 & 1 & 1 & 1 \\ -1 & -2x-1 & 1 & -1 \end{pmatrix} \partial + \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -2 \end{pmatrix}. \tag{28}$$

Applying the proposed algorithm in Theorem 1 to matrix differential operator (28) similar to example 1, one can construct two unimodular matrix differential operators  $S$  and  $T$  as

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix},$$

$$T = \begin{pmatrix} 1 & 0 & \frac{1-x\partial}{x\partial+x+1} & \frac{x\partial+\partial+x}{x\partial+x+1} \\ 0 & 1 & \frac{2\partial+1}{\partial(x\partial+x+1)} & \frac{\partial^2-1}{\partial(x\partial+x+1)} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{29}$$

such that the given system is in reduced form as follows:

$$\tilde{L} = \begin{pmatrix} \partial & (1+x)\partial & 0 & 0 \\ 1 & -x\partial & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{30}$$

where  $\tilde{L} = SLT$  has the form  $\begin{pmatrix} \tilde{L}^{11} & 0 \\ 0 & 0 \end{pmatrix}$  and

$$\tilde{L}^{11} = \begin{pmatrix} \partial & (1+x)\partial \\ 1 & -x\partial \end{pmatrix}, \tag{31}$$

is invertible matrix differential operator.

*Example 3.* Consider the following system of differential-algebraic equations to verify that the reduced system and the given system of DAEs have the same solution:

$$\begin{aligned} u_1' + u_3' + u_3 &= 0, \\ u_1 + 2u_2' + u_3 &= x, \\ 2u_1' + 2u_3' + 2u_3 &= 0. \end{aligned} \tag{32}$$

The solution of the given system (32) is  $u_1 = c_1, u_2 = (x^2/2) + (c_2e^{-x}/2) - (c_1x/2) + c_3$  and  $u_3 = c_2e^{-x}$ . In particular, if we take  $u_1 = 0$ , then the solution becomes  $u_1 = 0, u_2 = (x^2/2) + (c_2e^{-x}/2) + c_2$  and  $u_3 = c_1e^{-x}$ .

The operator notation  $Lu = f$  of the given system (14) is given by

$$\begin{pmatrix} \partial & 0 & \partial+1 \\ 1 & 2\partial & 1 \\ 2\partial & 0 & 2\partial+1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 0 \\ x \\ 0 \end{pmatrix}, \tag{33}$$

where  $L = \begin{pmatrix} \partial & 0 & \partial+1 \\ 1 & 2\partial & 1 \\ 2\partial & 0 & 2\partial+1 \end{pmatrix}$ . Now, apply the proposed algorithm to matrix differential operator  $L$  to get a reduced

operator  $\tilde{L}$  with two unimodular matrix differential operators  $S$  and  $T$ . We get

$$\tilde{L} = \begin{pmatrix} \partial & 0 & 0 \\ 1 & 2\partial & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}, \tag{34}$$

$$T = \begin{pmatrix} 1 & 0 & \frac{1+\partial}{\partial} \\ 0 & 1 & \frac{1}{2\partial^2} \\ 0 & 0 & 1 \end{pmatrix}.$$

System (33) is reduced to  $\tilde{L}(z(x)) = \tilde{f}(x)$ , where  $u(x) = T(z(x))$  and  $\tilde{f}(x) = S(f(x))$ . On simplification, we can get  $z(x) = T^{-1}(u(x))$  and  $\tilde{f}(x) = S(f(x))$ .

Therefore, we have

$$\tilde{L}(z(x)) = \tilde{f}(x) \implies \tilde{L}(T^{-1}(u(x))) = S(f(x))$$

$$\begin{aligned} &\implies \begin{pmatrix} \partial & 0 & 0 \\ 1 & 2\partial & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & \frac{1+\partial}{\partial} \\ 0 & 1 & \frac{1}{2\partial^2} \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ x \\ 0 \end{pmatrix} \end{aligned} \tag{35}$$

$$\begin{aligned} &\implies \begin{pmatrix} \partial & 0 & 0 \\ 1 & 2\partial & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 + \frac{\partial+1}{\partial}u_3 \\ u_2 - \frac{u_3}{2\partial^2} \\ u_3 \end{pmatrix} = \begin{pmatrix} 0 \\ x \\ 0 \end{pmatrix} \\ &\implies \begin{pmatrix} \partial u_1 + (\partial+1)u_3 \\ u_1 + 2\partial u_2 + u_3 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ x \\ 0 \end{pmatrix}. \end{aligned}$$

Now the reduced system of DAEs is

$$\begin{aligned} u_1' + u_3' + u_3 &= 0, \\ u_1 + 2u_2' + u_3 &= x. \end{aligned} \tag{36}$$

Solution of the reduced system (36), for  $u_1 = 0$ , is  $u_1 = 0$ ,  $u_2 = (x^2/2) + (c_1 e^{-x}/2) + c_2$  and  $u_3 = c_1 e^{-x}$ .

One can observe that the solution of the given system of DAEs (32) and the reduced system of DAEs (36) have the same solution. We can also observe that solving the reduced system (36) (contains two equations only) is simple compared to solving the given system (32) (contains three equations).

### 3. Maple Implementation

In this section, we discuss the Maple implementation of the algorithm by creating different data types. Using the Maple package, one can obtain the two unimodular matrix differential operators  $S, T$  and the reduced matrix differential operator of the given system. In Maple implementation,  $x$  is complex variable and  $\delta = (d/dx)$  is the differential operator.

**Input:**  $\mathcal{A}$  and  $\mathcal{B}$ , the coefficient matrices of a given matrix differential operator  $L = \mathcal{A}\delta + \mathcal{B}$ .

**Output:**  $\tilde{L}$ ,  $S$ , and  $T$ , the reduced matrix differential operator  $\tilde{L}$  of a given matrix differential operator  $L$  and two unimodular matrix differential operators  $S$  and  $T$ .

#### 3.1. Pseudocode

- (1)  $A, B \leftarrow$  coefficient matrices
- (2)  $m \leftarrow$  row dimension of  $L$
- (3)  $n \leftarrow$  column dimension of  $L$
- (4)  $r \leftarrow$  rank of  $A$
- (5)  $k \leftarrow$  rank of  $B$
- (6)  $NS\_Ls \leftarrow$  left null space of  $L$
- (7)  $S_1 \leftarrow$  identity matrix with  $LNS\_Ls$  as bottom block matrix
- (8)  $NS\_Lt \leftarrow$  right null space of  $L$
- (9)  $T_1 \leftarrow$  identity matrix with  $NS\_Lt$  as right block matrix
- (10)  $L_1 \leftarrow S_1.L.T_1$
- (11)  $A_1, B_1 \leftarrow$  coefficient matrices of  $L_1$
- (12)  $NS\_A_1 \leftarrow$  left null space of  $A_1$
- (13)  $S_2 \leftarrow$  identity matrix with  $LNS\_A_1$  as bottom block matrix
- (14)  $L_2 \leftarrow S_2.L_1$
- (15)  $A_2, B_2 \leftarrow$  coefficient matrices of  $L_2$
- (16)  $NS\_B_2 \leftarrow$  right null space of  $B_2$
- (17)  $T_2 \leftarrow$  identity matrix with  $NS\_B_2$  as right block matrix
- (18)  $L_3 \leftarrow L_2.T_2$
- (19)  $S \leftarrow S_2.S_1$
- (20)  $T \leftarrow T_1.T_2$

3.2. *Maple Code.* Using the procedure `ArrangeZeroLines`, we can replace the zero rows at the bottom and zero columns at right side of the matrix.

```

ArrangeZeroLines := proc (M::Matrix)
local m,n,ZR,L1,DR,m1,ZM,A1,ZC,L2,DC,m2,ZM2,
A2;
uses LinearAlgebra;
m,n := op(1, M);
ZR := LinearAlgebra: -ZeroVector[row](n);
L1 := [seq('if(LinearAlgebra: -Equal(M[i],ZR),i,
NULL), i = 1..m)];
DR := LinearAlgebra: -DeleteRow(M, L1);
m1 := op([1, 1], DR);
ZM := LinearAlgebra: -ZeroMatrix(m - m1, n);
A1:= convert(linalg[blockmatrix](2, 1, [DR, ZM]),
Matrix);
ZC:= LinearAlgebra: -ZeroVector(m);
L2 := [seq('if(LinearAlgebra: -Equal(A1
[()..(),i],ZC),i,NULL), i = 1..n)];
DC := LinearAlgebra: -DeleteColumn(A1, L2);
m2 := op([1, 2], DC);
ZM2 := LinearAlgebra: -ZeroMatrix(m, n - m2);
A2 := linalg[blockmatrix](1, 2, [DC, ZM2]);
return convert(A2, Matrix);
end proc;

```

The following procedure is `DAEs_Reduction`: the reduced matrix differential operator of a given matrix differential operator with two unimodular matrix differential operators. In this procedure,  $\delta = (d/dz)$  is differential operator and  $x$  is complex variable.

```

DAEs_Reduction := proc (A::Matrix, B::Matrix)
local L,m,n,r,k,Id,transpose_L,NS_Ls,Id_partA,
NS_partA,
S1,NS_Lt,Id_part3,NS_part3,T1,L1na,L1,A1,B1,
transpose_A1,NS_A1,Id_partA1,NS_partA1,S2,L2na,L2,
A2,B2,NS_B2,Id_partB2,NS_partB2,T2,L3na,
L3,A3,B3,S,T;
uses MatrixPolynomialAlgebra;
L := A * delta + B;
m := LinearAlgebra: -RowDimension(L);
n := LinearAlgebra: -ColumnDimension(L);
r := MTM: -rank(A);
k := MTM: -rank(B);
Id := LinearAlgebra: -IdentityMatrix(n);
transpose_L := LinearAlgebra: -Transpose(L);
NS_Ls := LinearAlgebra: -NullSpace(transpose_L);
Id_partA := LinearAlgebra: -Transpose
(Matrix(' ~ '[LinearAlgebra: -Transpose](' ~ '
[convert]

```

```

([seq(Id[i],i=1..m-nops(NS_Ls)),Matrix]));
NS_partA := LinearAlgebra: -Transpose (Matrix(' ~ '
[convert]
([seq(NS_Ls[i], i=1 .. Nops(NS_Ls)), Matrix]));
S1 := convert(linalg: -blockmatrix(2,1,[Id_partA,
NS_partA]),Matrix);
NS_Lt := ' ~ '[simplify](LinearAlgebra: -NullSpace(L));
Id_part3 := Matrix(' ~ '[LinearAlgebra: -Transpose]
(' ~ '[convert]
([seq(Id[i], i=1 .. n-nops(NS_Lt)), Matrix]));
NS_part3 := Matrix(' ~ '[convert]
([seq(NS_Lt[i], i=1 .. Nops(NS_Lt)), Matrix));
T1 := convert(linalg: -blockmatrix(1,2,[Id_part3,
NS_part3]),Matrix);
L1na := ' ~ '[simplify](S1.L.T1);
L1 := ArrangeZeroLines(L1na);
A1:= seq(MatrixPolynomialAlgebra: -Coeff(L1,delta,i),i=
0..1)[2];
B1 := seq(MatrixPolynomialAlgebra: -Coeff(L1,delta,
i),i=0..1)[1];
transpose_A1:= LinearAlgebra: -Transpose(A1);
NS_A1:= LinearAlgebra: -NullSpace(transpose_A1);
Id_partA1:=LinearAlgebra: -Transpose(Matrix
(' ~ '[LinearAlgebra: -Transpose](' ~ '[convert]
([seq(Id[i], i=1 .. m-nops(NS_A1)), Matrix]));
NS_partA1 := LinearAlgebra: -Transpose(Matrix(' ~ '
[convert]
([seq(NS_A1[i], i=1 .. Nops(NS_A1)), Matrix]));
S2 := convert(linalg: -blockmatrix(2,1,[Id_partA1,NS_
partA1]),Matrix);
L2na := ' ~ '[simplify](S2.L1);
L2 := ArrangeZeroLines(L2na);
A2 := seq(MatrixPolynomialAlgebra: -Coeff(L2,delta,
i),i=0..1)[2];
B2 := seq(MatrixPolynomialAlgebra: -Coeff(L2,delta,
i),i=0..1)[1];
NS_B2 := ' ~ '[simplify](LinearAlgebra: -NullSpace
(B2));
Id_partB2 := Matrix(' ~ '[LinearAlgebra: -Transpose]
(' ~ '[convert]([seq(Id
[i],i= nops(NS_B2)+1..n)),Matrix));
NS_partB2 := Matrix(' ~ '[convert]
([seq(NS_B2[i], i=1..nops(NS_B2)), Matrix));
T2 := LinearAlgebra: -IdentityMatrix(n);
L3na := ' ~ '[simplify](L2.T2);
L3 := ArrangeZeroLines(L3na);

```

```

A3 := seq(MatrixPolynomialAlgebra: -Coeff(L3,delta,
i),i=0..1)[2];
B3 := seq(MatrixPolynomialAlgebra: -Coeff(L3,delta,
i),i=0..1)[1];
S := ' ~ '[simplify](S2.S1);
T := ' ~ '[simplify](T1.T2);
return S, T, L3
end proc;

```

3.3. Sample Computations

Example 4. Consider the following matrix differential operator as given in Example 2:

$$L = \begin{pmatrix} \partial & (1+x)\partial & 1 & \partial+1 \\ 1 & -x\partial & \partial & -1 \\ \partial+1 & \partial & \partial+1 & \partial \\ -\partial+1 & (-2x-1)\partial & \partial-1 & -\partial-2 \end{pmatrix}, \tag{37}$$

where  $\mathcal{A} = \begin{pmatrix} 1 & x+1 & 0 & 1 \\ 0 & -x & 1 & 0 \\ 1 & 1 & 1 & 1 \\ -1 & -2x-1 & 1 & -1 \end{pmatrix}$  and

$\mathcal{B} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -2 \end{pmatrix}$ . Using Maple implementation, with

DAEs\_Reduction in the proposed algorithm, we have the following:

```

>A := Matrix([[1, x+1, 0, 1], [0, -x, 1, 0], [1, 1, 1, 1],
[-1, -2x-1, 1, -1]]);
>B := Matrix([[0, 0, 1, 1], [1, 0, 0, -1], [1, 0, 1, 0], [1,
0,-1, -2]]);

```

$$A := \begin{bmatrix} 1 & x+1 & 0 & 1 \\ 0 & -x & 1 & 0 \\ 1 & 1 & 1 & 1 \\ -1 & -2x-1 & 1 & -1 \end{bmatrix}, \tag{38}$$

$$B := \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -2 \end{bmatrix}$$

```

>S, T, Lred := DAEs_Reduction(A, B);

```



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 & \frac{\delta x - 1}{\delta x + x + 1} & \frac{\delta x + \delta + x}{\delta x + x + 1} \\ 0 & 1 & \frac{2\delta + 1}{\delta(\delta x + x + 1)} & \frac{\delta^2 - 1}{\delta(\delta x + x + 1)} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{39}$$

$$\begin{bmatrix} \delta & \delta(1+x) & 0 & 0 \\ 1 & -\delta x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

From Maple implementation, we have that two unimodular matrices differential operators  $S, T$  are

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix},$$

$$T = \begin{pmatrix} 1 & 0 & \frac{1-x\partial}{x\partial+x+1} & \frac{x\partial+\partial+x}{x\partial+x+1} \\ 0 & 1 & \frac{2\partial+1}{\partial(x\partial+x+1)} & \frac{\partial^2-1}{\partial(x\partial+x+1)} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{40}$$

and the reduced matrix differential operator of the given DAS (37) is

$$\tilde{L} = \begin{pmatrix} \partial & (1+x)\partial & 0 & 0 \\ 1 & -x\partial & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{41}$$

One can also verify in Maple that  $\tilde{L} = SLT$  as follows:

$$\begin{aligned} >L := \text{Matrix}([\delta, \delta * (1+x), 1, \delta + 1], \\ & [1, -\delta * x, \delta, -1], [\delta + 1, \delta, \delta + 1, \delta], \\ & \delta], \end{aligned}$$

$$[-\delta + 1, -\delta * (1 + 2 * x), \delta - 1, -\delta - 2]]);$$

$$\begin{bmatrix} \delta & \delta(1+x) & 1 & \delta + 1 \\ 1 & -\delta x & \delta & -1 \\ \delta + 1 & \delta & \delta + 1 & \delta \\ -\delta + 1 & -\delta(1+2x) & \delta - 1 & -\delta - 2 \end{bmatrix}. \tag{42}$$

>simplify ~ (S.L.T);

$$\begin{bmatrix} \delta & \delta(1+x) & 0 & 0 \\ 1 & -\delta x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{43}$$

### 4. Conclusion

In this paper, we discussed a new reduction algorithm to solve a given system of linear differential-algebraic equations with power series coefficients. With the proposed algorithm, one can transform the given matrix differential operator into another simple equivalent matrix differential operator using the elementary algebraic techniques. Certain examples are presented to illustrate the algorithm. The implemented Maple package is discussed and sample computations are presented.

### Data Availability

The datasets generated and analyzed during the current study are included within the article.

### Additional Points

In this paper, the authors presented a new reduction algorithm to solve a system of linear DAEs with power series coefficients. In this algorithm, they transform the given matrix differential operator to simple equivalent matrix differential operator using the elementary algebraic techniques. Several examples are presented to illustrate the algorithm and also the Maple package is discussed with sample computations.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### References

- [1] M. Günther and P. Rentrop, "The differential-algebraic index concept in electric circuit simulation," *Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 76, no. 1, pp. 91-94, 1996.
- [2] M. Günther and U. Feldmann, "The DAE-index in electric circuit simulation," *Mathematics and Computers in Simulation*, vol. 39, no. 5-6, pp. 573-582, 1995.

- [3] S. Thota and S. D. Kumar, "Solving system of higher-order linear differential equations on the level of operators," *International Journal of Pure and Applied Mathematics*, vol. 106, no. 1, pp. 11–21, 2016.
- [4] D. E. Schwarz and C. Tischendorf, "Structural analysis of electric circuits and consequences for MNA," *International Journal of Circuit Theory Applications*, vol. 28, no. 2, pp. 131–162, 2000.
- [5] P. Rabier and W. C. Rheinboldt, *Nonholonomic Motion of Rigid Mechanical Systems from a DAE Viewpoint*, p. 148, SIAM, Philadelphia, PA, USA, 2000.
- [6] B. Simeon, C. Fuhrer, and P. Rentrop, "Differential-algebraic equations in vehicle system dynamics," *Surveys on Mathematics for Industry*, vol. 1, pp. 1–37, 1991.
- [7] R. Gani and I. T. Cameron, "Modelling for dynamic simulation of chemical processes: the index problem," *Chemical Engineering Science*, vol. 47, no. 5, pp. 1311–1315, 1992.
- [8] C. C. Pantelides, D. Gritsis, K. R. Morison, and R. W. H. Sargent, "The mathematical modelling of transient systems using differential-algebraic equations," *Computers & Chemical Engineering*, vol. 12, no. 5, pp. 449–454, 1988.
- [9] J. Unger, A. Kröner, and W. Marquardt, "Structural analysis of differential-algebraic equation systems-theory and applications," *Computers & Chemical Engineering*, vol. 19, no. 8, pp. 867–882, 1995.
- [10] K. E. Brennan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, p. 268, North-Holland, Amsterdam, Netherlands, 1989.
- [11] J. W. Ponton and P. J. Gawthrop, "Systematic construction of dynamic models for phase equilibrium processes," *Computers & Chemical Engineering*, vol. 15, no. 12, pp. 803–808, 1991.
- [12] E. Griepentrog and R. Marz, *Differential-Algebraic Equations and Their Numerical Treatment*, p. 220, Teubner, Leipzig, Germany, 1986.
- [13] E. Hairer, C. Lubich, and M. Roche, "The numerical solution of differential-algebraic systems by Runge-Kutta methods," *Lecture Notes in Math*, p. 1409, Springer, Berlin, Germany, 1989.
- [14] S. Thota and S. D. Kumar, "A new method for general solution of system of higher-order linear differential equations," in *Proceedings of the International Conference on Interdisciplinary Research in Engineering and Technology*, pp. 240–243, Delhi, India, April 2015.
- [15] S. Thota and S. D. Kumar, "Symbolic algorithm for a system of differential-algebraic equations," *Kyungpook Mathematical Journal*, vol. 56, no. 4, pp. 1141–1160, 2016.
- [16] S. Thota, "On Maple implementation of a reduction algorithm for differential-algebraic systems," in *Proceedings of the International Conference on Mathematical Advances & Applications 2020*, Istanbul, Turkey, November 2020.
- [17] C. Shi, "Linear differential-algebraic equations of higher-order and the regularity or singularity of matrix polynomials," Ph.D. thesis, Von der Fakultät II—Mathematik und Naturwissenschaften der Technischen Universität, Berlin, Germany, 2004.
- [18] C. W. Gear, "Differential-algebraic equation index transformations," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, no. 1, pp. 39–47, 1988.
- [19] M. P. Quéré-Stuchlik, "Algorithmique des faisceaux linéaires de matrices, applications à la théorie des systèmes linéaires et à la résolution d'équations algébro-différentielles," Ph.D. thesis, 1996.
- [20] M. P. Quéré-Stuchlik and G. Villard, "An algorithm for the reduction of linear DAE," in *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation, ISSAC'95*, pp. 223–231, ACM Press, Montreal, Canada, April 1995.
- [21] P. Kunkel and V. Mehrmann, "Canonical forms for linear differential-algebraic equations with variable coefficients," *Journal of Computational and Applied Mathematics*, vol. 56, no. 3, pp. 225–251, 1994.
- [22] S. Schulz, "Four lectures on differential algebraic equations," Technical Report 497, The University of Auckland, Auckland, New Zealand, 2003.
- [23] C. El Bachaa, "Méthodes algébriques pour la résolution d'équations différentielles matricielles d'ordre arbitraire," Ph.D. thesis, Université de Limoges, Limoges, France, 2011.
- [24] V. Mehrmann and C. Shi, "Transformation of high order linear differential-algebraic systems to first order," *Numerical Algorithms*, vol. 42, no. 3–4, pp. 281–307, 2006.
- [25] S. Thota, "Initial value problems for system of differential-algebraic equations in Maple," *BMC Research Notes*, vol. 11, p. 651, 2018.
- [26] M. Barkatou, C. El Bacha, and E. Pflügel, "Simultaneously row- and column-reduced higher-order linear differential systems," in *Proceedings of ISSAC'10*, pp. 45–52, ACM, Munich, Germany, July 2010.
- [27] B. Beckermann, H. Cheng, and G. Labahn, "Fraction-free row reduction of matrices of ore polynomials," *Journal of Symbolic Computation*, vol. 41, no. 5, pp. 513–543, 2006.
- [28] T. Kailath, *Linear Systems*, Prentice-Hall, Upper Saddle River, NJ, USA, 1980.
- [29] W. A. Harris, Y. Sibuya, and L. Weinberg, "A reduction algorithm for linear differential systems," *Funkcialaj Ekvacioj*, vol. 11, pp. 59–67, 1968.